REMARKS

Claims 1-15 and 17-21 are pending. Claim 22 is new. Claims 1-15 and 17-21 stand rejected.

Regarding the Rejections under 35 U.S.C. §103

Claims 1-15 and 17-21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Chow et al (US 6,292,589) in view of Whitted et al, "A software Testbed for the Development of 3D Raster Graphics Systems" (January 1982, ACM Transactions on Graphics, New York, NY, USA Volume 1, Issue 1, Pages 43-58, hereafter "Whitted") in further view of Fuchs et al, "Pixel Planes 5: A Heterogeneous Multiprocessor Graphics System Using Processor-Enhanced Memories", ACM, Computer Graphics, Volume 23, no. 3, July 1989, pages 79-88 (hereinafter, "Fuchs"). These rejections are respectfully traversed.

Regarding claims 1, 7, 11, 13 and 15, claim 1 recites that "said multiplying a column or row of pixels with a predetermined matrix to generate a corresponding set of output pixels, determining, and sampling the pixels are performed by a shader module," claim 7 recites that "said creating a polyline and creating a line are performed by a shader module," claim 11 recites that "said multiplying and adding are performed by a shader module," and claim 13 recites that "for each column in a block of pixels, setting up a shader and rendering a scanline, and for each row in a block of pixels, setting up a shader and rendering a column, and wherein the setting up and the rendering are performed by a shader module." Additionally, Claim 15 recites "for each set of scanlines, sampling the pixels comprised within the scanlines and multiplying the sampled pixels with a row or column of the predetermined matrix, and wherein said setting up and the rendering are performed by a shader module." The Office Action admits that the Chow et al reference does not teach calculation or pixel manipulation within a shader module and looks to the Whitted and Fuchs references to remedy this deficiency.

The Office Action appears to assert that the Fuchs reference discloses "processing each block of pixels, in parallel, within at least one shader module" at the abstract and at page 81, Section 4, however, it does not. The disclosure in the abstract of Fuchs provides a

description of a hardware-based rendering machine with a software calculator to perform the rendering calculations on groups of pixels. On page 81, the Fuchs reference further discloses that "[o]nce transformed, sorted, and stored, a new scene is rendered by assigning all available Renderers to patches on the screen and dispatching to these Renderers primitives from their corresponding bins. When a Renderer completes a patch, it can discard its z-buffer and all other pixel values besides colors; pixel color values are transferred from on-chip pixel memory to the secondary storage system... The Renderer is then assigned to the next patch to be processed." There is no disclosure of a shader module of any type in either the abstract or page 81 of the Fuchs reference. A Renderer and a shader module are related graphics functions, however, a renderer and a shader module are employed for different functions. A rendering function is applied to calculate the effect of light sources on color and intensity of pixels within a graphics frame. A shader module calculates the 3-D surface properties of objects in terms of effects, colors, textures and shapes to generate complex, realistic scenes, and may perform this function singly, without accessing a separate rendering function. Thus, the disclosure of a rendering process in Fuchs does not disclose at least the "processing… within at least one shader module" feature as recited in independent claims 1, 7, 11, 13 and 15.

The Office Action also appears to assert that the Whitted reference remedies the deficiency in the Chow reference by disclosing in Figure 1, page 44 of that reference that generating "a corresponding set of output pixels, determining, and sampling the pixels are performed by said at least one shader module," however, it does not.

The instant invention is directed to a shader module used in association with a discrete cosine transformation (DCT) algorithm to process incoming 8x8 blocks of pixels by forming a cross product with an orthogonal matrix to preserve the values of the incoming pixels quickly in block and matrix dot product sets. These sets are preserved as textured lines and scanlines and stored into memory quickly, thus eliminating pixel loss. The input video data capture process is therefore greatly enhanced. To process this incoming data prior to delivery to a CPU or an output process, a shader module is employed using the correct matrix values and array offsets to allow DCT processing of each group of textured lines and scanlines as parallel processes. This process improves not only data capture but also improves processing speed for all pixels within the captured video data.

The recitations in the claims are not simply in reference to a shader, but to what processing is occurring within the shader as a part of the overall process. Claim 1, for example, recites "processing each block within at least one shader module" where "said multiplying a column or row of pixels with a predetermined matrix to generate a corresponding set of output pixels, determining, and sampling the pixels are performed by a shader module," which details the offloading of the CPU by performing these actions by a shader module within the GPU. The Office Action seems to assert that the inclusion of a shader function within Figure 1 of the Whitted reference teaches that these functions are performed by the shader function disclosed in that figure, however, it does not.

The Whitted reference includes a shader function that performs an interpolation of intensity values for pixels processed using a z-buffer to determine visibility (page 48, last paragraph). This is not the same as performing matrix operations such as multiplying and sampling within a shader module as recited in the claims. The Whitted reference further discloses on page 51, third full paragraph, that operations are performed by other arithmetic modules in order to minimize "the number of unnecessary operations performed by the shader". In Whitted, the shader is reserved for the final display stage in order to determine pixel intensity values only (page 48, last full paragraph and page 51, first full paragraph). There is no disclosure or teaching of a shader module within a GPU performing matrix operations such as "multiplying a column or row of pixels with a predetermined matrix to generate a corresponding set of output pixels, determining, and sampling the pixels" as recited in the submitted claim 1.

In addition, the Office Action seems to assert that "the skilled artisan could have combined and/or substituted the method of processing the blocks of pixels as taught by Fuchs with the method of performing that processing in a shader module as taught by Whitted" to obtain the features of the independent claims, however, this analysis is in error. A rendering function, as disclosed in Fuchs, is not the same as a shader module. As shown above, there is no disclosure of a shader module in Fuchs, which means that the calculations for surface effects, textures, and shapes for which a graphics engine relies upon a shader module cannot be performed. As shown above, the Whitted reference does not disclose matrix operations within a shader module or sampling the pixels within the shader module, reserving a shader for a determination of pixel intensity values only. Thus, the disclosure in Fuchs does not

provide the ability to perform the calculations necessary to the performance of shader functionality, and Whitted does not disclose the new features within a shader as disclosed in the independent claims.

The combination of the Chow, Fuchs and Whitted references does not teach or disclose "multiplying a column or row of pixels with a predetermined matrix to generate a corresponding set of output pixels, determining, and sampling the pixels" as recited in the submitted claim 1, "said creating a polyline and creating a line are performed by a shader module," as recited in claim 7, "said multiplying and adding are performed by a shader module" as recited in claim 11, "setting up a shader and rendering a scanline, and for each row in a block of pixels, setting up a shader and rendering a column" as recited in claim 13, or "sampling the pixels comprised within the scanlines and multiplying the sampled pixels with a row or column of the predetermined matrix, and wherein said setting up and the rendering are performed by a shader module" as recited in claim 15. Therefore, the combination of Chow et al. and Whitted fails to provide the teachings needed to establish that claims 1, 7, 11, 13 and 15 are obvious. These claims are allowable for at least the reasons given above. Accordingly, reconsideration and allowance are respectfully requested. New claim 22 incorporates many of the foregoing limitations and thus, it is patentable over the cited references for the same reasons.

Regarding claims 2-6, 8-10, 12, 14-15, and 17-21, these claims each depend from one of independent claims 1, 7, 11, 13 or 15. In view of the above, it is clear that the combination of Chow et al. Fuchs and Whitted fails to provide the teachings to establish that claims 1, 7, 11, 13 and 15 are obvious. The dependant claims are, therefore, allowable for at least the reasons shown for claims 1, 7, 11, 13 and 15. Nevertheless, the applicants wish to point out additional reasons why dependent claims 4 and 19 are patentable over the cited art.

Regarding claim 4, this claim recites "sampling the pixels comprised within the scanlines comprises using a separate shader for each set of scanlines." The Office Action seems to assert that this feature is disclosed in the Chow et al. reference at Col 10, line 18 and in Figures 6(a) – 6(c), however, it is not. The Chow et al. reference in Figures 6(a)-6(c) and in Col 10 in its entirety are completely silent with regard to shader operations of any type.

Chow et al. in Col 10 at lines 18-36 discloses macroblock matching techniques for video frames, and that these matching techniques may be accomplished by using a series of adder circuits. This in no way discloses the recited claim element. This disclosure is for a matching technique, not a video processing function such as sampling scanlines, and this implementation does not provide for the use of shaders at all. Therefore, the Chow et al. reference does not provide the disclosure or teaching to render claim 4 obvious. Reconsideration and allowance are respectfully requested.

Regarding claim 19, this claim recites "the GPU defines an array of coordinate offsets to neighboring pixels, wherein the shader accesses the pixels in the scanlines using the offset array." The Office Action seems to assert that this feature is disclosed in the Chow et al. reference at Col 6, line 18, however, it is not. Chow et al. at Col 6, lines 18-24 discloses that scanlines for video arrive at different resolution levels for NTSC or PAL formats and that the pixel data arrives as a stream of scanlines. There is no disclosure of either a shader for accessing and processing the incoming stream of pixels, or that there is any type of offset into the data required to access the scanline pixels, as recited in the claim. Therefore, the Chow et al. reference does not provide the disclosure or teaching to render claim 19 obvious. Accordingly, reconsideration and allowance are respectfully requested.

CONCLUSION

For the foregoing reasons, Applicants respectfully submit that the instant application is in condition for allowance.

Date: August 28, 2008

/Steven B. Samuels/
Steven B. Samuels
Registration No. 37,711

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439